DOI: 10.3901/CJME.2016.1026.127, available online at www.springerlink.com; www.cjmenet.com

Quaternion-based Nonlinear Trajectory Tracking Control of a Quadrotor Unmanned Aerial Vehicle

ZHA Changliu, DING Xilun*, YU Yushu, and WANG Xueqiang

School of Mechanical Engineering and Automation, Beihang University, Beijing 100191, China

Received June 14, 2016; revised October 8, 2016; accepted October 26, 2016

Abstract: At present, most controllers of quadrotor unmanned aerial vehicles(UAVs) use Euler angles to express attitude. These controllers suffer a singularity problem when the pitch angle is near 90°, which limits the maneuverability of the UAV. To overcome this problem, based on the quaternion attitude representation, a 6 degree of freedom(DOF) nonlinear controller of a quadrotor UAV is designed using the trajectory linearization control(TLC) method. The overall controller contains a position sub-controller and an attitude sub-controller. The two controllers regulate the translational and rotational motion of the UAV, respectively. The controller is improved by using the commanded value instead of the nominal value as the input of the inner control loop. The performance of controller is tested by simulation before and after the improvement, the results show that the improved controller is better. The proposed controller is also tested via numerical simulation and real flights and is compared with the traditional controller based on Euler angles. The test results confirm the feasibility and the robustness of the proposed nonlinear controller. The proposed controller can successfully solve the singularity problem that usually occurs in the current attitude control of UAV and it is easy to be realized.

Keywords: unmanned aerial vehicle, quaternion, nonlinear control, trajectory linearization control, singularity

1 Introduction

In recent years, micro unmanned aerial vehicles(UAVs) with stable hovering flight and vertical take-off and landing(VTOL) capabilities have gained in popularity among researchers and hobbyists. They are appropriate for many applications, such as intelligence, rescue, surveillance, reconnaissance and various other similar missions. The quadrotor UAV is a four-rotor aerial robot with VTOL ability. The translational and rotational motion of a quadrotor UAV is realized by varying the speed of the four rotors. Compared with a traditional helicopter, the quadrotor UAV does not contain complex mechanisms. Its major advantages are that it offers a larger payload and is simpler to operate.

The quadrotor UAV is nevertheless an under-actuated, nonlinear, and strongly coupled system with 6 degrees of freedom(DOF). Consequently, the development of a reliable quadrotor UAV flight control system is a challenging task. Researchers have investigated many control techniques for quadrotor UAVs such as PID[1–2], slide mode[3–4], LQR[5–6] and back-stepping[7–8]. PID is widely used due to its simplicity[9]. And the slide mode controller is popular for it can be used to deal with the system model uncertainty[10]. These controllers can usually work in the conditions when the attitude of the aircraft varies smally. Currently, many research groups are focusing on improving the quadrotor UAV's ability to aggressively maneuver and interact with the environment. For example, MELLINGER, et al[11–12], studied how to control a quadrotor UAV to enable aggressive maneuvers such as flying through narrow, vertical gaps and perching on inverted surfaces with high precision and repeatability. YU, et al[13], and DING, et al[14], proposed a multi-propeller multifunction aerial robot(MMAR) capable of flying, wall-climbing and arm-operating. These maneuvers require a higher performance controller for quadrotor UAVs. However, most of the controllers are designed based on the representation of the Euler angle. These controllers suffer from a singularity problem when the pitch angle of the UAV is near 90°. One method to overcome this problem is to use the exponential coordinate, and numerous studies have focused on exponential coordinate-based tracking control[15–16]. For example, YU, et al[17], proposed a exponential coordinated-based tracking controller for quadrotor. This type of method has been proven to be effective in theory. However, for experimental activity, it is necessary to design an observer to estimate the exponential coordinates from the angular velocities. Real-time implementation of this algorithm is challenging. Another method to avoid singularities is to use quaternions. LIU, et

* Corresponding author. E-mail: xlding@buaa.edu.cn
Supported by National Science Foundation for Distinguished Young Scholars of China(Grant No. 51125020), National Natural Science Foundation of China(Grant No.51505014), and China Postdoctoral Science Foundation(Grant No. 2016T90024)

al[18], proposed a quaternion-based attitude controller for quadrotor UAV and proved its stability. The quaternion attitude representation is usually applied to integrated navigation[19]. At present, UAV navigation systems can usually output the attitude in the form of a quaternion, which simplifies the task of designing a UAV controller based on the quaternion attitude representation. Compared with the Euler angle, the other advantage of a quaternion is that it requires only algebraic and not trigonometric operations, thus reducing the computational workload.

A quadrotor UAV is a nonlinear system; to improve the performance of the controller, a nonlinear control method is needed to design the controller. The trajectory linearization control(TLC) method, presented by HUANG, et al[20], is a novel nonlinear tracking and decoupling control technique. It combines nonlinear dynamic inversion and linear time-varying feedback stabilization in a novel way. Nonlinear tracking and decoupling control via TLC can be viewed as an ideal gain-schedule controller designed at every point along the trajectory. Thus, TLC provides robust stability without interpolation of control gains. More importantly, compared with the conventional gain scheduling controller, tuning the parameters of a controller designed via TLC is simple.

In this paper, based on the quaternion attitude representation, we present a new nonlinear trajectory tracking controller for a quadrotor UAV. A control model of the UAV based on the quaternion representation of attitude is derived. The attitude of the UAV is 3 dimensional(3D); however, the quaternion has 4 elements. Thus, a redundancy problem exists when using the quaternion to represent attitude. This problem is carefully solved in the design of the attitude controller. Moreover, the position and attitude controller, according to the quaternion-based model, is constructed using the TLC method.

This paper is structured as follows. The control model of the quadrotor UAV based on the quaternion is introduced in section 2. The TLC controller design, which is based on the derived model, is described in section 3. Section 4 presents several simulation and flight test results. Finally, conclusions are given in section 5.

2 Dynamic Model of a Quadrotor UAV

2.1 Introduction of quaternions

The quaternion attitude representation is a four-parameter representation based on the fact that any rotation of a rigid body can be described by a single rotation around a vector $\boldsymbol{\eta}$ that is defined with respect to the reference frame[21]. It is possible to define a quaternion as

$$\boldsymbol{\beta}=\begin{pmatrix}a\\b\\c\\d\end{pmatrix}=\begin{pmatrix}\cos\eta\\(\eta_x/\eta)\sin\eta\\(\eta_y/\eta)\sin\eta\\(\eta_z/\eta)\sin\eta\end{pmatrix}, \tag{1}$$

where $\boldsymbol{\beta}$ is a four-element vector and a, b, c and d are its four components, which are functions of the orientation of this vector and the magnitude of the rotation. The parameters η_x, η_y, η_z are the components of the angle vector $\boldsymbol{\eta}$, and η is the magnitude of $\boldsymbol{\eta}$. Unlike Euler angles, quaternion rotations can change their single axis continuously and therefore do not require a set of predefined rotation axes. This method of rotating around an arbitrary direction has only one axis of rotation and does not lose degrees of freedom; therefore, there are no singularities. The quaternion elements satisfy an equation of constraint, which is called the normality condition and is represented as follows:

$$a^2+b^2+c^2+d^2=1. \tag{2}$$

2.2 Model of the quadrotor using quaternions

Disregarding the aerodynamics and the gyroscopic effect induced by the propellers, the quadrotor UAV can be idealized as a 6-DOF rigid body. The rigid UAV motion can be decomposed into translational motions and rotational motions. The coordinate frames and the configuration of a rigid quadrotor UAV are illustrated in Fig. 1, where Σ_e is the earth-fixed frame and Σ_b is the body-fixed frame. The attitude of the body-fixed frame Σ_b to the earth-fixed frame Σ_e is defined by three Euler angles, denoted as roll, pitch, and yaw angle in the aircraft convention.

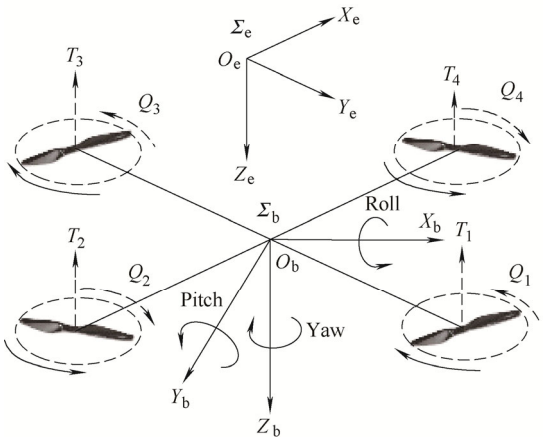

Fig. 1. Rigid quadrotor UAV configuration

The attitude of the UAV is the rotation of the body-fixed frame with respect to the earth-fixed frame; it can be represented by a rotation matrix based on a quaternion. The rotation matrix is represented as[22]

$$\boldsymbol{R}_b^e=\begin{pmatrix}a^2+b^2-c^2-d^2 & 2(bc-ad) & 2(bd+ac)\\ 2(bc+ad) & a^2-b^2+c^2-d^2 & 2(cd-ab)\\ 2(bd-ac) & 2(cd+ab) & a^2-b^2-c^2+d^2\end{pmatrix}, \tag{3}$$

where $\boldsymbol{R}_b^e$ represents the rotation matrix. The kinematics equation for translational motion is given by

$$\dot{\boldsymbol{P}} = \boldsymbol{v}, \tag{4}$$

where the position $\boldsymbol{P}$ and velocity $\boldsymbol{v}$ of the quadrotor UAV are described relative to the earth-fixed frame.

The dynamics equation for translational motion is

$$\dot{\boldsymbol{v}} = \frac{1}{m}\boldsymbol{F}, \tag{5}$$

where $\boldsymbol{F}$ and m represent the overall force of the UAV in the earth-fixed frame and the mass of the UAV, respectively.

The kinematic equation for rotational motion in a quaternion is given by

$$\dot{\boldsymbol{\beta}} = \boldsymbol{C}(\boldsymbol{\beta})\boldsymbol{\omega} = \frac{1}{2}\begin{pmatrix} -b & -c & -d \\ a & -d & c \\ d & a & -b \\ -c & b & a \end{pmatrix}\begin{pmatrix} p \\ q \\ r \end{pmatrix}, \tag{6}$$

where $\boldsymbol{\omega}=(p \quad q \quad r)^{\mathrm{T}}$ represents the body roll, pitch, and yaw velocity, respectively, of the UAV in the body-fixed frame.

The dynamic equation for rotational motion can be expressed as follows[20]:

$$\dot{\boldsymbol{\omega}} = \boldsymbol{J}^{-1}(\boldsymbol{M} - \boldsymbol{\omega}\times\boldsymbol{J}\boldsymbol{\omega}) = \begin{pmatrix} J^p_{pq}pq + J^p_{qr}qr \\ J^q_{pp}p^2 + J^q_{rr}r^2 + J^q_{pr}pr \\ J^r_{pq}pq + J^r_{qr}qr \end{pmatrix} + \begin{pmatrix} g^p_l & 0 & g^p_n \\ 0 & g^q_m & 0 \\ g^r_l & 0 & g^r_n \end{pmatrix}\boldsymbol{M}, \tag{7}$$

where $\boldsymbol{M}$ and $\boldsymbol{J}$ represent the control torque and the inertia tensor with respect to the body-fixed frame attached to the mass center of the UAV, respectively; the coefficients J^*_{**} and g^*_* are defined as the mass moments of inertia of the airframe. Due to the quadrotor UAV symmetry, J_{xy}=J_{yz}=0; thus, J^*_{**} and g^*_* with J_{xx}, J_{yy}, J_{zz}, and J_{xz} can be expressed as[23]

$$\begin{cases} J^p_{pq} = J_{xz}(J_{yy} - J_{zz} - J_{xx})D = -J^r_{qr}, \\ J^p_{qr} = (J_{zz}{}^2 - J_{yy}J_{zz} - J_{xx}{}^2)D, \\ J^q_{pp} = -\dfrac{J_{xz}}{J_{yy}} = -J^q_{rr}, \\ J^q_{pp} = \dfrac{J_{zz} - J_{xx}}{J_{yy}}, \\ J^r_{pq} = (J_{xx}J_{yy} - J_{xz}{}^2 - J_{xx}{}^2)D, \\ g^p_l = -J_{zz}D, \\ g^p_n = -J_{xz}D = g^r_l, \\ g^q_m = \dfrac{1}{J_{yy}}, \\ g^r_n = -J_{xx}D, \\ D = \dfrac{1}{J_{xz}{}^2 - J_{xx}J_{zz}}. \end{cases} \tag{8}$$

The torque $\boldsymbol{M}$ and force $\boldsymbol{F}$ are used as virtual controls in Eqs. (5) and (7). A quadrotor UAV is actuated by four identical propellers, as shown in Fig. 1. The thrust T_i and the reactive torque Q_i generated by the propeller i can be expressed as[24]

$$T_i = k_{\mathrm{t}}\Omega_i^2, \quad Q_i = k_{\mathrm{d}}\Omega_i^2, \tag{9}$$

where Ω_i represents the rotational speed of the propeller i, and k_{t} and k_{d} are proportional parameters that depend on the density of air and the shape, size, and pitch angle of the blades, as well as other factors[25].

The total thrust $T = \sum_{i=1}^{4} T_i$ and the body moment $\boldsymbol{M}_{\mathrm{a}}$ generated by the propellers' force and torque can be expressed as

$$\begin{pmatrix} \boldsymbol{M}_{\mathrm{a}} \\ T \end{pmatrix} = \boldsymbol{L}\begin{pmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{pmatrix} = \begin{pmatrix} -\dfrac{\sqrt{2}}{4}d_{\mathrm{r}}k_{\mathrm{t}} & -\dfrac{\sqrt{2}}{4}d_{\mathrm{r}}k_{\mathrm{t}} & \dfrac{\sqrt{2}}{4}d_{\mathrm{r}}k_{\mathrm{t}} & \dfrac{\sqrt{2}}{4}d_{\mathrm{r}}k_{\mathrm{t}} \\ \dfrac{\sqrt{2}}{4}d_{\mathrm{r}}k_{\mathrm{t}} & -\dfrac{\sqrt{2}}{4}d_{\mathrm{r}}k_{\mathrm{t}} & -\dfrac{\sqrt{2}}{4}d_{\mathrm{r}}k_{\mathrm{t}} & \dfrac{\sqrt{2}}{4}d_{\mathrm{r}}k_{\mathrm{t}} \\ -k_{\mathrm{d}} & k_{\mathrm{d}} & -k_{\mathrm{d}} & k_{\mathrm{d}} \\ k_{\mathrm{t}} & k_{\mathrm{t}} & k_{\mathrm{t}} & k_{\mathrm{t}} \end{pmatrix}\begin{pmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{pmatrix}, \tag{10}$$

where d_{r} represents the distance between the diagonal pair of propellers. The gyroscopic effects induced by the rotational motors and propellers are given by[24]

$$\boldsymbol{M}_{\mathrm{G}} = \sum_{i=1}^{4} J_{\mathrm{r}}(\boldsymbol{\omega}\times\boldsymbol{e}_z)(-1)^i\Omega_i, \tag{11}$$

where J_{r} is the moment of inertia of one motor-propeller and $\boldsymbol{e}_z$=(0, 0, 1)$^{\mathrm{T}}$. Thus, the total moment $\boldsymbol{M}$ can be expressed as

$$\boldsymbol{M} = \boldsymbol{M}_{\mathrm{a}} - \boldsymbol{M}_{\mathrm{G}}. \tag{12}$$

The propellers are in a body-fixed configuration; only a vertical propulsive force T can be generated. Thus, the total forces $\boldsymbol{F}$ can be expressed in the earth-fixed frame as

$$\boldsymbol{F} = \boldsymbol{R}^{\mathrm{e}}_{\mathrm{b}}\begin{pmatrix} 0 \\ 0 \\ -T \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} - \begin{pmatrix} \mathrm{sgn}(v_x)\mathrm{C}_{Dv}v_x^2 \\ \mathrm{sgn}(v_y)\mathrm{C}_{Dv}v_y^2 \\ \mathrm{sgn}(v_z)\mathrm{C}_{Dv}v_z^2 \end{pmatrix}, \tag{13}$$

where $\mathrm{sgn}(v) = \begin{cases} +1 & \text{if } v > 0, \\ -1 & \text{otherwise}, \end{cases}$

and C_{Dv}>0 is the drag force coefficient due to velocity in

units of N • m^2/s^2. The velocities v_x, v_y and v_z are expressed in the earth-fixed frame, and g and m represent the gravity constant and mass of the UAV, respectively.

3 TLC Design Using Quaternions

The control model of a rigid quadrotor UAV based on quaternions is given in Eqs. (4)–(13). In this section, we design the position and attitude controller of a rigid body based on this model using the TLC method.

3.1 TLC controller structure

Here, we briefly introduce the TLC method used to design the position and attitude tracking controller[20, 26].

The nonlinear dynamic system can be expressed as follows:

$$\begin{cases}\dot{\boldsymbol{\xi}}(t)=f\left(\boldsymbol{\xi}(t),\boldsymbol{\mu}(t)\right),\\ \boldsymbol{\eta}(t)=h\left(\boldsymbol{\xi}(t),\boldsymbol{\mu}(t)\right),\end{cases} \tag{14}$$

where $\boldsymbol{\xi}(t)\in \mathrm{R}^n$, $\boldsymbol{\mu}(t)\in \mathrm{R}^l$, $\boldsymbol{\eta}(t)\in \mathrm{R}^m$ are the state, input and output, respectively, and the mappings $f(\cdot,\cdot)$: $\mathrm{R}^n\times\mathrm{R}^l\to\mathrm{R}^n$ and $h(\cdot,\cdot):\mathrm{R}^n\times\mathrm{R}^l\to\mathrm{R}^m$ are bounded and locally Lipschitz, respectively. Let $\bar{\boldsymbol{\xi}}(t)$, $\bar{\boldsymbol{\mu}}(t)$, $\bar{\boldsymbol{\eta}}(t)$ be the nominal state, nominal control, and nominal output trajectories, respectively, satisfying

$$\begin{cases}\dot{\bar{\boldsymbol{\xi}}}(t)=f\left(\bar{\boldsymbol{\xi}}(t),\bar{\boldsymbol{\mu}}(t)\right),\\ \bar{\boldsymbol{\eta}}(t)=h\left(\bar{\boldsymbol{\xi}}(t),\bar{\boldsymbol{\mu}}(t)\right).\end{cases} \tag{15}$$

The tracking errors are defined by

$$\begin{cases}\tilde{\boldsymbol{\xi}}(t)=\boldsymbol{\xi}(t)-\bar{\boldsymbol{\xi}}(t),\\ \tilde{\boldsymbol{\mu}}(t)=\boldsymbol{\mu}(t)-\bar{\boldsymbol{\mu}}(t),\\ \tilde{\boldsymbol{\eta}}(t)=\boldsymbol{\eta}(t)-\bar{\boldsymbol{\eta}}(t),\end{cases} \tag{16}$$

where $\tilde{\boldsymbol{\xi}}(t)$, $\tilde{\boldsymbol{\mu}}(t)$ and $\tilde{\boldsymbol{\eta}}(t)$ represent the state, input and output tracking error of the UAV, respectively. The state and output tracking error dynamics can be written as

$$\begin{cases}\dot{\tilde{\boldsymbol{\xi}}}(t)=F\left(\tilde{\boldsymbol{\xi}},\tilde{\boldsymbol{\mu}},\bar{\boldsymbol{\xi}},\bar{\boldsymbol{\mu}}\right)=\\ \qquad f\left(\bar{\boldsymbol{\xi}}(t)+\tilde{\boldsymbol{\xi}}(t),\bar{\boldsymbol{\mu}}(t)+\tilde{\boldsymbol{\mu}}(t)\right)-f\left(\bar{\boldsymbol{\xi}}(t),\bar{\boldsymbol{\mu}}(t)\right),\\ \tilde{\boldsymbol{\eta}}(t)=H\left(\tilde{\boldsymbol{\xi}},\tilde{\boldsymbol{\mu}},\bar{\boldsymbol{\xi}},\bar{\boldsymbol{\mu}}\right)=\\ \qquad h\left(\bar{\boldsymbol{\xi}}(t)+\tilde{\boldsymbol{\xi}}(t),\bar{\boldsymbol{\mu}}(t)+\tilde{\boldsymbol{\mu}}(t)\right)-h\left(\bar{\boldsymbol{\xi}}(t),\bar{\boldsymbol{\mu}}(t)\right).\end{cases} \tag{17}$$

As shown in Fig. 2, the control consists of two parts: an open-loop dynamic inverse I/O mapping of the plant to compute the nominal control function $\bar{\boldsymbol{\mu}}(t)$ for any given nominal output trajectory $\bar{\boldsymbol{\eta}}(t)$ and a closed-loop tracking error stabilizing control law $\tilde{\boldsymbol{\mu}}\left(\tilde{\boldsymbol{\xi}},\bar{\boldsymbol{\xi}}\right)$ to account for modeling simplifications, uncertainties, disturbances and excitations of internal dynamics.

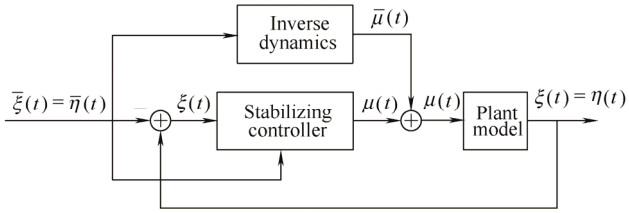

Fig. 2. TLC controller

The TLC method has been successfully applied to several types of vehicles[27–28]. In these applications, all control loops of the controller are designed according to the nominal value, except the first control loop. The tracking control law of the outer control loop is not transmitted to the dynamic inversion of the next control loop. Therefore, the dynamic inversion of the inner control loop does not have any correction effect on the outer control tracking error; the position control allocation must calculate separately the nominal attitude command and corrective attitude command.

Based on the above analysis, the control structure and robustness are improved using the commanded value that is corrected by the tracking error control law instead of the nominal value as the input of the dynamic inversion of the inner control loop to suppress the error, together with the stabilizing controller. Another advantage of the improved control structure is that the position control allocation calculates the attitude command by using a force command directly, which simplifies the design of the controller. The improved controller structure is shown in Fig. 3. The controller consists of two parts, namely, position control and attitude control, which control the translational motion and rotational motion of the UAV, respectively.

3.2 Attitude controller

We start our design with the attitude controller. As shown in Fig. 3, the attitude controller consists of an outer loop and an inner loop, which control the attitude and body rate of the quadrotor UAV, respectively. The attitude controller receives the attitude command and total thrust force command from the position controller and calculates the actuator speed command for the UAV.

3.2.1 Outer loop control

The outer loop of the attitude controller is designed according to the rotational kinematics of Eq. (6), but the equation cannot be inverted directly. We can rewrite this equation as

$$\dot{\boldsymbol{\beta}}=\boldsymbol{C}(\boldsymbol{\beta})\boldsymbol{\omega}=\boldsymbol{H}(\boldsymbol{\beta})\boldsymbol{\tau}=\\ \frac{1}{2}\begin{pmatrix}a & -b & -c & -d\\ b & a & -d & c\\ c & d & a & -b\\ d & -c & b & a\end{pmatrix}\begin{pmatrix}0\\ p\\ q\\ r\end{pmatrix}, \tag{18}$$

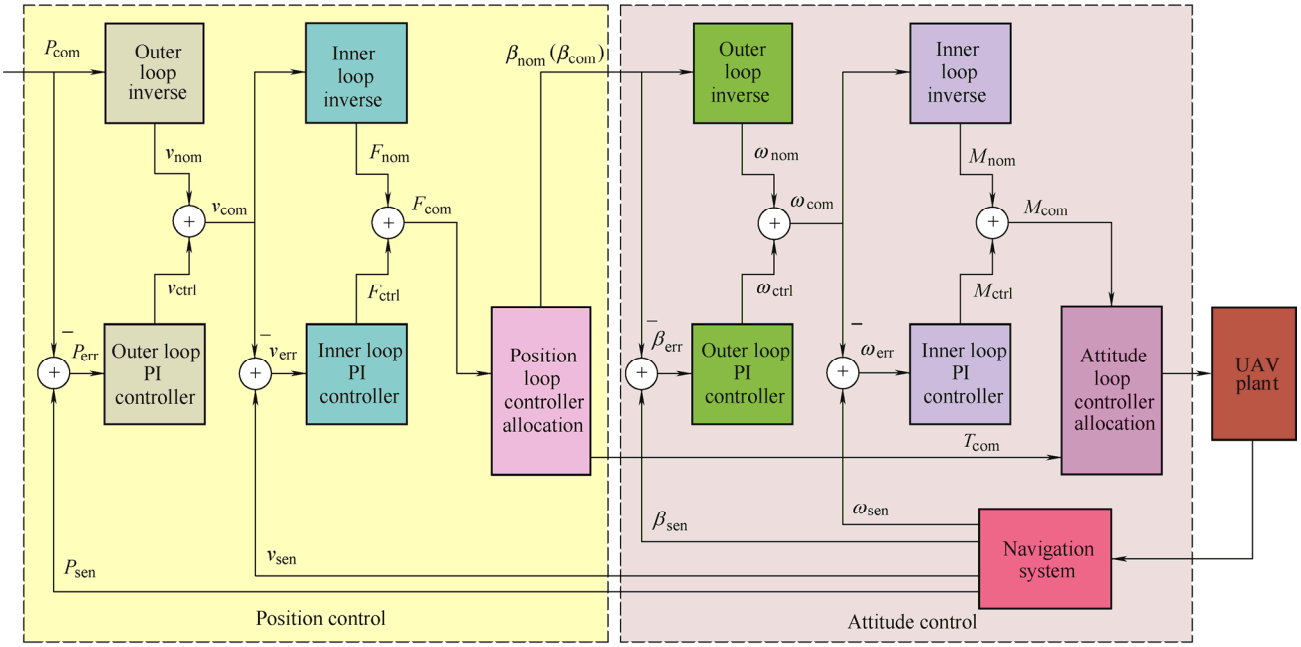

Fig. 3. Six-DOF trajectory linearization controller block diagram

where $\boldsymbol{\tau}=\left(0,\ \boldsymbol{\omega}^{\mathrm{T}}\right)^{\mathrm{T}}$. The nominal body angular velocities are obtained by inverting Eq. (18) as follows:

$$\omega_{\mathrm{nom}}=2\begin{pmatrix}-b_{\mathrm{nom}} & a_{\mathrm{nom}} & d_{\mathrm{nom}} & -c_{\mathrm{nom}}\\ -c_{\mathrm{nom}} & -d_{\mathrm{nom}} & a_{\mathrm{nom}} & b_{\mathrm{nom}}\\ -d_{\mathrm{nom}} & c_{\mathrm{nom}} & -b_{\mathrm{nom}} & a_{\mathrm{nom}}\end{pmatrix}\begin{pmatrix}\dot{a}_{\mathrm{nom}}\\ \dot{b}_{\mathrm{nom}}\\ \dot{c}_{\mathrm{nom}}\\ \dot{d}_{\mathrm{nom}}\end{pmatrix},\quad (19)$$

where $\dot{a}_{\mathrm{nom}}$, $\dot{b}_{\mathrm{nom}}$, $\dot{c}_{\mathrm{nom}}$, $\dot{d}_{\mathrm{nom}}$ are calculated from a_{nom}, b_{nom}, c_{nom}, d_{nom} using the second-order pseudo-differentiator with a transfer-function[20], i.e.,

$$G_{\mathrm{d}}(s)=\frac{\omega_{\mathrm{n,d}}^2 s}{s^2+2\zeta_{\mathrm{d}}\omega_{\mathrm{n,d}}s+\omega_{\mathrm{n,d}}^2}.\quad (20)$$

The error dynamics of the loop are given by

$$\dot{\boldsymbol{\beta}}_{\mathrm{err}}=C(\boldsymbol{\beta}_{\mathrm{nom}})\boldsymbol{\omega}_{\mathrm{ctrl}}.\quad (21)$$

The attitude error is defined as the difference between the sensed values and the command values, i.e.,

$$\boldsymbol{\beta}_{\mathrm{err}}=\boldsymbol{\beta}_{\mathrm{sen}}-\boldsymbol{\beta}_{\mathrm{com}}.\quad (22)$$

The feedback control law for the loop is given by

$$\boldsymbol{\omega}_{\mathrm{ctrl}}=-\boldsymbol{K}_{\mathrm{P3}}(t)\boldsymbol{\beta}_{\mathrm{err}}-\boldsymbol{K}_{\mathrm{I3}}(t)\int\boldsymbol{\beta}_{\mathrm{err}}.\quad (23)$$

For the outer loop of the attitude controller, we augment the state variables with the integral of the quaternion and define the control input and state variable vector as

$$\boldsymbol{\beta}_{\mathrm{aug}}=\begin{pmatrix}\int\boldsymbol{\beta}\mathrm{d}t\\ \boldsymbol{\beta}\end{pmatrix}\in\mathrm{R}^8,\quad \boldsymbol{\mu}_3=\boldsymbol{\tau}.\quad (24)$$

From Eq. (24), the augmented nonlinear state space model can be expressed as follows:

$$\dot{\boldsymbol{\beta}}_{\mathrm{aug}}=f_3(\boldsymbol{\beta}_{\mathrm{aug}},\boldsymbol{\mu}_3)=\begin{pmatrix}\boldsymbol{\beta}\\ \boldsymbol{H}(\boldsymbol{\beta})\boldsymbol{\tau}\end{pmatrix}.\quad (25)$$

Via the TLC method, the augmented linearized tracking error system can be expressed as follows:

$$\ddot{\tilde{\boldsymbol{\beta}}}_{\mathrm{aug}}=\boldsymbol{A}_3\tilde{\boldsymbol{\beta}}_{\mathrm{aug}}+\boldsymbol{B}_3\tilde{\boldsymbol{\tau}},\quad (26)$$

where $\tilde{\boldsymbol{\tau}}$ is the tracking error control law for the outer loop and $\tilde{\boldsymbol{\beta}}_{\mathrm{aug}}$ is a vector containing the tracking error state variables, as follows:

$$\tilde{\boldsymbol{\beta}}_{\mathrm{aug}}=\begin{pmatrix}\int(\boldsymbol{\beta}-\boldsymbol{\beta}_{\mathrm{com}})\mathrm{d}t\\ \boldsymbol{\beta}-\boldsymbol{\beta}_{\mathrm{com}}\end{pmatrix}.\quad (27)$$

The Jacobian matrices $\boldsymbol{A}_3$ in Eq. (26) are expressed as follows:

$$\boldsymbol{A}_3=\frac{\partial f_3}{\partial\boldsymbol{\beta}_{\mathrm{aug}}}\Big|_{\bar{\boldsymbol{\beta}}_{\mathrm{aug}},\bar{\boldsymbol{\tau}}}=\left(\begin{array}{c|c}\boldsymbol{0}_4 & \boldsymbol{E}_4\\ \hline \boldsymbol{0}_4 & A_{322}\end{array}\right),\quad (28)$$

where $\boldsymbol{O}_4$ is a 4×4 null matrix, $\boldsymbol{E}_4$ denotes a 4×4 identity matrix, and $\boldsymbol{A}_{322}$ is given by

$$A_{322}=\left(\frac{\partial(\boldsymbol{H}(\boldsymbol{\beta})\boldsymbol{\tau})}{\partial\boldsymbol{\beta}}\Big|_{\boldsymbol{\beta}_{\mathrm{nom}},\boldsymbol{\omega}_{\mathrm{nom}}}\right)=$$

$$\frac{1}{2}\begin{pmatrix} 0 & -p_{\text{nom}} & -q_{\text{nom}} & -r_{\text{nom}} \\ p_{\text{nom}} & 0 & r_{\text{nom}} & -q_{\text{nom}} \\ q_{\text{nom}} & -r_{\text{nom}} & 0 & p_{\text{nom}} \\ r_{\text{nom}} & q_{\text{nom}} & -p_{\text{nom}} & 0 \end{pmatrix}. \tag{29}$$

The Jacobian matrices $\boldsymbol{B}_3$ in Eq. (26) are expressed as follows:

$$\boldsymbol{B}_3 = \frac{\partial f_3}{\partial \boldsymbol{\omega}}\Big|_{\bar{\boldsymbol{\beta}}_{\text{aug}},\bar{\tau}} = \begin{pmatrix} \boldsymbol{0}_{4\times 4} \\ \hline \boldsymbol{B}_{321} \end{pmatrix}, \tag{30}$$

where $\boldsymbol{O}_4$ is a 4×4 null matrix and $\boldsymbol{B}_{321}$ is given by

$$\boldsymbol{B}_{321} = \left(\frac{\partial(\boldsymbol{H}(\boldsymbol{\beta})\boldsymbol{\tau})}{\partial \boldsymbol{\tau}}\Big|_{\boldsymbol{\beta}_{\text{nom}},\boldsymbol{\omega}_{\text{nom}}}\right) = \boldsymbol{H}(\boldsymbol{\beta}_{\text{nom}}) = \frac{1}{2}\begin{pmatrix} a_{\text{nom}} & -b_{\text{nom}} & -c_{\text{nom}} & -d_{\text{nom}} \\ b_{\text{nom}} & a_{\text{nom}} & -d_{\text{nom}} & c_{\text{nom}} \\ c_{\text{nom}} & d_{\text{nom}} & a_{\text{nom}} & -b_{\text{nom}} \\ d_{\text{nom}} & -c_{\text{nom}} & b_{\text{nom}} & a_{\text{nom}} \end{pmatrix}. \tag{31}$$

For outer loop attitude quaternion errors, the state feedback control law is designed as

$$\tilde{\boldsymbol{\tau}} = -\boldsymbol{K}_3\tilde{\boldsymbol{\beta}}_{\text{aug}}, \tag{32}$$

where $\boldsymbol{K}_3 = \begin{pmatrix} \boldsymbol{K}_{\text{I3}} & \boldsymbol{K}_{\text{P3}} \end{pmatrix}$. Thus, the closed-loop system matrix can be determined in decoupled multi-variable phase-variable canonical form as follows:

$$\boldsymbol{A}_{\text{cl3}} = \begin{pmatrix} \boldsymbol{O}_4 & E_4 \\ \hline \boldsymbol{A}_{\text{cl31}} & \boldsymbol{A}_{\text{cl32}} \end{pmatrix}, \tag{33}$$

where $\boldsymbol{A}_{\text{cl31}}$=diag($-\alpha_{3j1}$), $\boldsymbol{A}_{\text{cl32}}$=diag($-\alpha_{3j2}$) and the coefficients α_{3jk}(j=0, 1, 2, 3; k=1, 2) are obtained from the closed-loop characteristic equation with the desired damping ratio and bandwidth. According to the theory of PD-eigenstructure[26], the coefficients can be expressed as

$$\alpha_{3j1}(t) = \omega_{\text{n},j}^2(t), \quad \alpha_{3j2}(t) = 2\varsigma_j\omega_{\text{n},j}(t) - \frac{\dot{\omega}_{\text{n},j}(t)}{\omega_{\text{n},j}(t)}, \tag{34}$$

where $\omega_{\text{n},j}(t)$ is the natural frequency and ς_j is the damping ratio; j=0, 1, 2, 3 represents the four channels. From $\boldsymbol{\mu}_3 = \boldsymbol{\tau} = \begin{pmatrix} 0, & \boldsymbol{\omega}^{\text{T}} \end{pmatrix}^{\text{T}}$, we know that the augmented linearized tracking error system actually has only three channels; the first channel does not exist. Thus, α_{301} and α_{302} can take any value. $\boldsymbol{A}_{\text{cl31}}$ and $\boldsymbol{A}_{\text{cl32}}$ can be rewritten as $\boldsymbol{A}_{\text{cl31}}$=diag(0, $-\alpha_{3j1}$) and $\boldsymbol{A}_{\text{cl32}}$=diag(0, $-\alpha_{3j2}$), respectively, where j=1, 2, 3 represent the 3 attitude channels.

Based on (26)–(33), we can obtain $\boldsymbol{A}_3-\boldsymbol{B}_3\boldsymbol{K}_3=\boldsymbol{A}_{\text{cl3}}$ and express the gain matrices as follows:

$$\boldsymbol{K}_{\text{P3}}(t) = \boldsymbol{H}^{-1}(\boldsymbol{\beta}_{\text{nom}})(\boldsymbol{A}_{322} - \boldsymbol{A}_{\text{cl32}}) = \begin{pmatrix} \boldsymbol{k}_{\text{p31}} & \boldsymbol{k}_{\text{p32}} & \boldsymbol{k}_{\text{p33}} & \boldsymbol{k}_{\text{p34}} \end{pmatrix}, \tag{35}$$

where

$$\boldsymbol{k}_{\text{p31}} = \begin{pmatrix} b_{\text{nom}}p_{\text{nom}} + c_{\text{nom}}q_{\text{nom}} + d_{\text{nom}}r_{\text{nom}} \\ a_{\text{nom}}p_{\text{nom}} - c_{\text{nom}}r_{\text{nom}} + d_{\text{nom}}q_{\text{nom}} \\ a_{\text{nom}}q_{\text{nom}} + b_{\text{nom}}r_{\text{nom}} - d_{\text{nom}}p_{\text{nom}} \\ a_{\text{nom}}r_{\text{nom}} - b_{\text{nom}}q_{\text{nom}} + c_{\text{nom}}p_{\text{nom}} \end{pmatrix},$$

$$\boldsymbol{k}_{\text{p32}} = \begin{pmatrix} 2b_{\text{nom}}\alpha_{312} - a_{\text{nom}}p_{\text{nom}} - c_{\text{nom}}r_{\text{nom}} + d_{\text{nom}}q_{\text{nom}} \\ 2a_{\text{nom}}\alpha_{312} + b_{\text{nom}}p_{\text{nom}} - c_{\text{nom}}q_{\text{nom}} - d_{\text{nom}}r_{\text{nom}} \\ b_{\text{nom}}q_{\text{nom}} - a_{\text{nom}}r_{\text{nom}} - 2d_{\text{nom}}\alpha_{312} + c_{\text{nom}}p_{\text{nom}} \\ 2c_{\text{nom}}\alpha_{312} + a_{\text{nom}}q_{\text{nom}} + b_{\text{nom}}r_{\text{nom}} + d_{\text{nom}}p_{\text{nom}} \end{pmatrix},$$

$$\boldsymbol{k}_{\text{p33}} = \begin{pmatrix} 2c_{\text{nom}}a_{322} - a_{\text{nom}}q_{\text{nom}} + b_{\text{nom}}r_{\text{nom}} - d_{\text{nom}}p_{\text{nom}} \\ 2d_{\text{nom}}a_{322} + a_{\text{nom}}r_{nom} + b_{nom}q_{nom} + c_{\text{nom}}p_{\text{nom}} \\ 2a_{\text{nom}}a_{322} - b_{\text{nom}}p_{\text{nom}} + c_{nom}q_{nom} - d_{\text{nom}}r_{\text{nom}} \\ c_{\text{nom}}r_{\text{nom}} - a_{\text{nom}}p_{\text{nom}} - 2b_{nom}a_{322} + d_{\text{nom}}q_{\text{nom}} \end{pmatrix},$$

$$\boldsymbol{k}_{\text{p34}} = \begin{pmatrix} 2d_{\text{nom}}\alpha_{332} - a_{\text{nom}}r_{\text{nom}} - b_{\text{nom}}q_{\text{nom}} + c_{\text{nom}}p_{\text{nom}} \\ b_{\text{nom}}r_{\text{nom}} - a_{\text{nom}}q_{\text{nom}} - 2\alpha_{332}c_{\text{nom}} + d_{\text{nom}}p_{\text{nom}} \\ 2b_{\text{nom}}\alpha_{332} + a_{\text{nom}}p_{\text{nom}} + c_{\text{nom}}r_{\text{nom}} + d_{\text{nom}}q_{\text{nom}} \\ 2a_{\text{nom}}\alpha_{332} - b_{\text{nom}}p_{\text{nom}} - c_{\text{nom}}q_{\text{nom}} + d_{\text{nom}}r_{\text{nom}} \end{pmatrix}.$$

$$\boldsymbol{K}_{\text{I3}}(t) = -\boldsymbol{H}^{-1}(\boldsymbol{\beta}_{\text{nom}})\boldsymbol{A}_{cl31} = \begin{pmatrix} 0 & 2b_{\text{nom}}\alpha_{311} & 2c_{\text{nom}}\alpha_{321} & 2d_{\text{nom}}\alpha_{331} \\ 0 & 2a_{\text{nom}}\alpha_{311} & 2d_{\text{nom}}\alpha_{321} & -2c_{\text{nom}}\alpha_{331} \\ 0 & -2d_{\text{nom}}\alpha_{311} & 2a_{\text{nom}}\alpha_{321} & 2b_{\text{nom}}\alpha_{331} \\ 0 & 2c_{\text{nom}}\alpha_{311} & -2b_{\text{nom}}\alpha_{321} & 2a_{\text{nom}}\alpha_{331} \end{pmatrix}. \tag{36}$$

The output of the outer loop attitude control is given by

$$\boldsymbol{\omega}_{\text{com}} = \boldsymbol{\omega}_{\text{nom}} + \boldsymbol{\omega}_{\text{ctrl}}, \tag{37}$$

where $\boldsymbol{\omega}_{\text{com}}$ is the commanded body angular velocity vector, which is output to the inner loop.

3.2.2 *Inner loop control*

Because the design process of each control loop is similar, we will focus on the final design results. The inner loop of the attitude controller is designed according to Eq. (7), which describes the dynamics of the rotational motion. The nominal body torque is calculated by inverting the equation as

$$\boldsymbol{M}_{\text{nom}} = \begin{pmatrix} M_{x,\text{nom}} \\ M_{y,\text{nom}} \\ M_{z,\text{nom}} \end{pmatrix} = \begin{pmatrix} J_{xx}\dot{p}_{\text{com}} + (J_{zz} - J_{yy})q_{\text{com}}r_{\text{com}} - J_{xz}(\dot{r}_{\text{com}} + q_{\text{com}}p_{\text{com}}) \\ J_{yy}\dot{q}_{\text{com}} + (J_{xx} - J_{zz})p_{\text{com}}r_{\text{com}} + J_{xz}(p_{\text{com}}^2 - r_{\text{com}}^2) \\ J_{zz}\dot{r}_{\text{com}} + (J_{yy} - J_{xx})q_{\text{com}}p_{\text{com}} + J_{xz}(q_{\text{com}}r_{\text{com}} - \dot{p}_{\text{com}}) \end{pmatrix}, \tag{38}$$

where $\dot{p}_{\text{com}}$, $\dot{q}_{\text{com}}$, $\dot{r}_{\text{com}}$ are calculated from p_{com}, q_{com}, r_{com}, respectively, using the second-order pseudo-differentiator in Eq. (20).

The PI feedback control law for the attitude inner loop is given by

$$\boldsymbol{M}_{\text{ctrl}} = -\boldsymbol{K}_{\text{P4}}(t)\boldsymbol{\omega}_{\text{err}} - \boldsymbol{K}_{\text{I4}}(t)\int \boldsymbol{\omega}_{\text{err}}, \tag{39}$$

where the body angular velocity tracking error $\boldsymbol{\omega}_{\text{err}} = \boldsymbol{\omega}_{\text{sen}} - \boldsymbol{\omega}_{\text{com}}$ and the inner loop PI feedback control gain matrices are expressed as

$$\boldsymbol{K}_{\text{P4}}(t) = \begin{pmatrix} \boldsymbol{k}_{\text{p41}} & \boldsymbol{k}_{\text{p42}} & \boldsymbol{k}_{\text{p43}} \end{pmatrix}, \tag{40}$$

where

$$\boldsymbol{k}_{\text{p41}} = \begin{pmatrix} J_{xx}(J^{p}_{pq}q_{\text{com}} + \alpha_{412}) - J_{xz}J^{r}_{pq}q_{\text{com}} \\ J_{yy}(2J^{p}_{pp}p_{\text{com}} + J^{q}_{pr}r_{\text{com}}) \\ -J_{xz}(J^{p}_{pq}q_{\text{com}} + \alpha_{412}) + J_{zz}J^{r}_{pq}q_{\text{com}} \end{pmatrix},$$

$$\boldsymbol{k}_{\text{p42}} = \begin{pmatrix} J_{xx}(J^{p}_{pq}p_{\text{com}} + J^{p}_{qr}r_{\text{com}}) - J_{xz}(J^{r}_{pq}p_{\text{com}} + J^{r}_{qr}r_{\text{com}}) \\ J_{yy}\alpha_{422} \\ -J_{xz}(J^{p}_{pq}p_{\text{com}} + J^{p}_{qr}r_{\text{com}}) + J_{zz}(J^{r}_{pq}p_{\text{com}} + J^{r}_{qr}r_{\text{com}}) \end{pmatrix},$$

$$\boldsymbol{k}_{\text{p43}} = \begin{pmatrix} J_{xx}J^{p}_{qr}q_{\text{com}} - J_{xz}(J^{r}_{qr}q_{\text{com}} + \alpha_{432}) \\ J_{yy}(2J^{q}_{rr}r_{\text{com}} + J^{q}_{pr}p_{\text{com}}) \\ -J_{xz}J^{p}_{qr}q_{\text{com}} + J_{zz}(J^{r}_{qr}q_{\text{com}} + \alpha_{432}) \end{pmatrix},$$

$$\boldsymbol{K}_{\text{I4}}(t) = \begin{pmatrix} J_{xx}\alpha_{411} & 0 & -J_{xz}\alpha_{431} \\ 0 & J_{yy}\alpha_{421} & 0 \\ -J_{xz}\alpha_{411} & 0 & J_{zz}\alpha_{431} \end{pmatrix}. \tag{41}$$

Then, the output of the inner loop attitude control is given by

$$\boldsymbol{M}_{\text{com}} = \boldsymbol{M}_{\text{nom}} + \boldsymbol{M}_{\text{ctrl}}, \tag{42}$$

where $\boldsymbol{M}_{\text{com}}$ is the commanded body torque vector, which is output to the attitude control allocation.

3.2.3 *Attitude control allocation*

Attitude control allocation is performed to calculate the actuator speed command for the quadrotor UAV. The speed command is calculated by inverting Eq. (10) as

$$\begin{pmatrix} \Omega^{2}_{1,\text{com}} \\ \Omega^{2}_{2,\text{com}} \\ \Omega^{2}_{3,\text{com}} \\ \Omega^{2}_{4,\text{com}} \end{pmatrix} = \boldsymbol{L}^{-1}\begin{pmatrix} \boldsymbol{M}_{\text{com}} + \boldsymbol{M}_{\text{G}} \\ T_{\text{com}} \end{pmatrix}. \tag{43}$$

This step completes the attitude controller design.

3.3 Position controller

We now present the design of the position controller. As shown in Fig. 3, the position controller consists of an outer loop and an inner loop, which control the position and velocity, respectively, of the quadrotor UAV. The position controller calculates the attitude command and total thrust force for the attitude controller by using the position command.

3.3.1 *Outer loop control*

As shown in the first loop from the left in Fig. 3, the outer loop of the position control consists of a dynamic inversion and a PI controller. The nominal velocity is obtained by inverting Eqs. (4), which describes the kinematics of the translational motion, as

$$\boldsymbol{v}_{\text{nom}} = \dot{\boldsymbol{P}}_{\text{nom}}, \tag{44}$$

where $\dot{\boldsymbol{P}}_{\text{nom}}$ is calculated from $\boldsymbol{P}_{\text{nom}}$ using the second-order pseudo-differentiator in Eq. (20).

The position error is defined as the difference between the sensed values and the command values, i.e.,

$$\boldsymbol{P}_{\text{err}} = \boldsymbol{P}_{\text{sen}} - \boldsymbol{P}_{\text{com}}. \tag{45}$$

The error dynamics of the outer-loop are given by

$$\dot{\boldsymbol{P}}_{\text{err}} = \boldsymbol{v}_{\text{ctrl}}. \tag{46}$$

The feedback control law for the outer loop is given by

$$\boldsymbol{v}_{\text{ctrl}} = -\boldsymbol{K}_{\text{P1}}(t)\boldsymbol{P}_{\text{err}} - \boldsymbol{K}_{\text{I1}}(t)\int \boldsymbol{P}_{\text{err}}. \tag{47}$$

The position loop PI feedback control gain matrices are given by

$$\boldsymbol{K}_{\text{P1}}(t) = -A_{\text{cl12}} = \begin{pmatrix} \alpha_{112} & 0 & 0 \\ 0 & \alpha_{122} & 0 \\ 0 & 0 & \alpha_{132} \end{pmatrix}, \tag{48}$$

$$\boldsymbol{K}_{\text{I1}}(t) = -A_{\text{cl11}} = \begin{pmatrix} \alpha_{111} & 0 & 0 \\ 0 & \alpha_{121} & 0 \\ 0 & 0 & \alpha_{131} \end{pmatrix}, \tag{49}$$

where $\boldsymbol{A}_{\text{cl11}}=\text{diag}\{-\alpha_{1j1}\}$, $\boldsymbol{A}_{\text{cl12}}=\text{diag}\{-\alpha_{1j2}\}$ and the coefficients $\alpha_{1jk}(j=1, 2, 3;\ k=1, 2)$ are the control parameters of the system. The output of the outer loop is the velocity command given by

$$\boldsymbol{v}_{\text{com}} = \boldsymbol{v}_{\text{nom}} + \boldsymbol{v}_{\text{ctrl}}, \tag{50}$$

where $\boldsymbol{v}_{\text{com}}$ is the commanded velocity vector, which is the input to the inner loop of the position controller.

3.3.2 Inner loop control

The inner loop of the position controller is designed according to Eq. (5), which describes the dynamics of the translational motion. The nominal force is calculated by inverting this equation as

$$\boldsymbol{F}_{\text{nom}}=\begin{pmatrix}F_{x,\text{nom}}\\F_{y,\text{nom}}\\F_{z,\text{nom}}\end{pmatrix}=m\begin{pmatrix}\dot{v}_{x,\text{com}}\\\dot{v}_{y,\text{com}}\\\dot{v}_{z,\text{com}}\end{pmatrix},\tag{51}$$

where $\dot{v}_{x,\text{com}}$, $\dot{v}_{y,\text{com}}$, $\dot{v}_{z,\text{com}}$ are calculated from $v_{x,\text{com}}$, $v_{y,\text{com}}$, $v_{z,\text{com}}$, respectively, using the second-order pseudo-differentiator in Eq. (20). We now define the error dynamics and design the stabilizing controller. The error dynamics of the earth-fixed frame velocity are given by

$$\dot{\boldsymbol{v}}_{\text{err}}=\frac{1}{m}\boldsymbol{F}_{\text{ctrl}},\tag{52}$$

where

$$\boldsymbol{v}_{\text{err}}=\boldsymbol{v}_{\text{sen}}-\boldsymbol{v}_{\text{com}}.\tag{53}$$

The inner loop feedback control is given by

$$\boldsymbol{F}_{\text{ctrl}}=-\boldsymbol{K}_{\text{P2}}(t)\boldsymbol{v}_{\text{err}}-\boldsymbol{K}_{\text{I2}}(t)\int\boldsymbol{v}_{\text{err}}.\tag{54}$$

The velocity loop gain matrices are given by

$$\boldsymbol{K}_{\text{P2}}(t)=-mA_{\text{cl22}}=m\begin{pmatrix}\alpha_{212}&0&0\\0&\alpha_{222}&0\\0&0&\alpha_{232}\end{pmatrix},\tag{55}$$

$$\boldsymbol{K}_{\text{I2}}(t)=-mA_{\text{cl21}}=m\begin{pmatrix}\alpha_{211}&0&0\\0&\alpha_{221}&0\\0&0&\alpha_{231}\end{pmatrix}.\tag{56}$$

The output of the inner loop is the force command given by

$$F_{\text{com}}=F_{\text{nom}}+F_{\text{ctrl}}.\tag{57}$$

3.3.3 Position control allocation

The position control allocation unit takes the force command as its input and calculates the attitude angles and the total thrust force along the body Z_{b} axis as its output. The attitude of the quadrotor UAV is the rotation of the body-fixed frame with respect to the earth-fixed frame, which can be represented by a direction cosine matrix, denoted by the symbol $\boldsymbol{R}_{\text{b}}^{\text{e}}$, as

$$\boldsymbol{R}_{\text{b}}^{\text{e}}=(\boldsymbol{r}_1\quad\boldsymbol{r}_2\quad\boldsymbol{r}_3)=\begin{pmatrix}c_{11}&c_{12}&c_{13}\\c_{21}&c_{22}&c_{23}\\c_{31}&c_{32}&c_{33}\end{pmatrix}.\tag{58}$$

From Eqs. (13) and (51), we can calculate $\boldsymbol{r}_3$ using

$$\boldsymbol{r}_3=-\frac{\boldsymbol{r}_0}{\|\boldsymbol{r}_0\|_2},\tag{59}$$

where $\boldsymbol{r}_0=\begin{pmatrix}F_x+\text{sgn}(v_x)\text{C}_{Dv}v_x^2\\F_y+\text{sgn}(v_y)\text{C}_{Dv}v_y^2\\F_z-mg+\text{sgn}(v_z)\text{C}_{Dv}v_z^2\end{pmatrix}$.

Note from Fig. 1 that the total force vector in the body-fixed frame does not depend on the yaw angle; thus, ψ can be assigned to any value. In practice, however, the yaw angle is assigned to the heading angle so that the X_{e} axis of the quadrotor UAV will always be aligned with the heading direction. The yaw angle ψ can be calculated by solving the following equation[20]:

$$\psi=\arctan\left(\frac{\dot{Y}_{\text{nom}}}{\dot{X}_{\text{nom}}}\right).\tag{60}$$

The column $\boldsymbol{r}_1$ of the rotation matrix projects onto the reference plane($X_{\text{e}}O_{\text{e}}Y_{\text{e}}$), i.e., the unit vector $\boldsymbol{a}$, can be expressed as

$$\boldsymbol{a}=(\cos\psi,\sin\psi,0)^{\text{T}}.\tag{61}$$

$\boldsymbol{r}_1$ can be derived according to the following relationship:

$$\boldsymbol{r}_1=\frac{\boldsymbol{b}\times\boldsymbol{r}_3}{\|\boldsymbol{b}\times\boldsymbol{r}_3\|_2}.\tag{62}$$

where $\boldsymbol{b}=\dfrac{\boldsymbol{e}_z\times\boldsymbol{a}}{\|\boldsymbol{e}_z\times\boldsymbol{a}\|_2}$, $\boldsymbol{e}_z$=(0, 0, 1)$^{\text{T}}$.

Because the direction cosine matrix is an orthogonal matrix, according to the right-hand rule, $\boldsymbol{r}_2$ can then be calculated as

$$\boldsymbol{r}_2=\boldsymbol{r}_3\times\boldsymbol{r}_1.\tag{63}$$

For small angular displacements, the quaternion parameters may be derived using the following relationships[21]:

$$\boldsymbol{\beta}_{\text{nom}}=\begin{pmatrix}a_{\text{nom}}\\b_{\text{nom}}\\c_{\text{nom}}\\d_{\text{nom}}\end{pmatrix}=\begin{pmatrix}\frac{1}{2}(c_{11}+c_{22}+c_{33})^{1/2}\\\frac{1}{4a_{\text{nom}}}(c_{32}-c_{23})\\\frac{1}{4a_{\text{nom}}}(c_{13}-c_{31})\\\frac{1}{4a_{\text{nom}}}(c_{21}-c_{12})\end{pmatrix}.\tag{64}$$

According to Fig. 3, the allocation unit does not include

the attitude error; thus, $\boldsymbol{\beta}_{\text{com}} = \boldsymbol{\beta}_{\text{nom}}$. A more comprehensive algorithm for the extraction of quaternion parameters from the direction cosines can be found elsewhere[29]. The total thrust force command T_{com} is sent directly to the actuator and can be obtained from Eq. (13) as

$$T_{\text{com}} = -\sqrt{F_x^2 + F_y^2 + F_z^2}\ , \tag{65}$$

where $F_x = F_{x,\text{com}} + \text{sgn}\left(v_{x,\text{com}}\right) C_{Dv} v_{x,\text{com}}^2$,

$F_y = F_{y,\text{com}} + \text{sgn}(v_{y,\text{com}}) C_{Dv} v_{y,\text{com}}^2$,

$F_z = F_{z,\text{com}} - mg + \text{sgn}\left(v_{z,\text{com}}\right) C_{Dv} v_{z,\text{com}}^2$.

We now have the complete quaternion-based nonlinear controller for the quadrotor UAV.

4 Test Results

4.1 Simulation results

To evaluate the performance of the proposed control method, we performed simulation tests using the MATLAB R2012b/SIMULINK environment. In the simulation, the quadrotor UAV parameters used are chosen as m=2.605 kg, J_r=22 895.55 g • mm^2, $\boldsymbol{J}$=diag(4.88×10^{-2}, 4.87×10^{-2}, 8.33×10^{-2})kg • m^2, k_d=8.02×10^{-7} Nm/(rad • s^{-1})^2, k_t= 2.63×10^{-5} N/(rad • s^{-1})^2, d_r=0.67 m, C_{Dv}=0.01 N • m^2/s^2, and g=9.8 m/s^2. Each quadrotor UAV motor-propeller is modeled as a two-order inertial system; the damping coefficient and the natural frequency of the inertial system are set to 0.707 and 30 rad/s, respectively[20]. The other coefficients chosen for the simulation are given in Table 1 and Table 2.

Table 1. Control coefficients in the position controller

Control loop	Channel number	Natural frequency ω_n/(rad • s^{-1})	Damping ratio ζ
Outer loop	X channel	0.05	1.414
	Y channel	0.05	1.414
	Z channel	0.05	2
Inner loop	X channel	0.2	1.414
	Y channel	0.2	1.414
	Z channel	0.2	2

Table 2. Control coefficients in the attitude controller

Control loop	Channel number	Natural frequency ω_n/(rad • s^{-1})	Damping ratio ζ
Outer loop	Roll channel	0.8	1.414
	Pitch channel	0.8	1.414
	Yaw channel	0.8	1.414
Inner loop	Roll channel	4.8	1.414
	Pitch channel	4.8	1.414
	Yaw channel	4.8	1.414

Three simulation test cases are presented in this paper: A) attitude tracking control based on quaternion and Euler angles, B) evaluation of robustness based on quaternion control, and C) comparison of the tracking performance of the traditional structure controller with that of the proposed controller.

4.1.1 *Case A*

To verify whether the proposed controller based on the quaternion can effectively eliminate the singular point associated with the Euler angle controller, simulation tests with a pitch angle of approximately 90° are carried out based on the Euler angle controller and the quaternion-based controller. For these tests, the position control loop is not considered; only the attitude loop is tested. The attitude angle command of the simulation input is $\phi_{\text{com}} = 0$, $\theta_{\text{com}} = \pi/2 + 0.2\sin(0.8t)$ and $\psi_{\text{com}} = 0$. The relationship between the attitude represented by the quaternion and the Euler angles is

$$\boldsymbol{\beta}_{\text{com}} = \begin{pmatrix} a_{\text{com}} \\ b_{\text{com}} \\ c_{\text{com}} \\ d_{\text{com}} \end{pmatrix} = \begin{pmatrix} \cos\left(\frac{\phi_{\text{com}}}{2}\right)\cos\left(\frac{\theta_{\text{com}}}{2}\right)\cos\left(\frac{\psi_{\text{com}}}{2}\right) + \sin\left(\frac{\phi_{\text{com}}}{2}\right)\sin\left(\frac{\theta_{\text{com}}}{2}\right)\sin\left(\frac{\psi_{\text{com}}}{2}\right) \\ \sin\left(\frac{\phi_{\text{com}}}{2}\right)\cos\left(\frac{\theta_{\text{com}}}{2}\right)\cos\left(\frac{\psi_{\text{com}}}{2}\right) - \cos\left(\frac{\phi_{\text{com}}}{2}\right)\sin\left(\frac{\theta_{\text{com}}}{2}\right)\sin\left(\frac{\psi_{\text{com}}}{2}\right) \\ \cos\left(\frac{\phi_{\text{com}}}{2}\right)\sin\left(\frac{\theta_{\text{com}}}{2}\right)\cos\left(\frac{\psi_{\text{com}}}{2}\right) + \sin\left(\frac{\phi_{\text{com}}}{2}\right)\cos\left(\frac{\theta_{\text{com}}}{2}\right)\sin\left(\frac{\psi_{\text{com}}}{2}\right) \\ \cos\left(\frac{\phi_{\text{com}}}{2}\right)\cos\left(\frac{\theta_{\text{com}}}{2}\right)\sin\left(\frac{\psi_{\text{com}}}{2}\right) + \sin\left(\frac{\phi_{\text{com}}}{2}\right)\sin\left(\frac{\theta_{\text{com}}}{2}\right)\cos\left(\frac{\psi_{\text{com}}}{2}\right) \end{pmatrix}. \tag{66}$$

From Eq. (66), we can obtain the attitude angle command $\boldsymbol{\beta}_{\text{com}}$ as represented by a quaternion. In this case, the force of gravity on the quadrotor UAV is almost parallel to the lift. The height cannot be maintained by the lift. Therefore, in these tests, the height of the UAV is not controlled. The initial values of the attitude and angular velocity are zero, the lift is a constant 28 N, and the test conditions for the two controllers are identical. The test results of the controller based on Euler angles are shown in Fig. 4. Because the pitch angle is near 90°, the controller based on Euler angles cannot realize effective control of the attitude due to the singular points.

The test results of the quaternion-based controller are shown in Figs. 5–9. From Fig. 5, it can be observed that the values of the four elements of the quaternion fluctuate with the commanded angle, although the changes are in a reasonable range, and there is no singular point. Fig. 6 shows the tracking effect of the pitch angle. The quaternion-based controller can still achieve effective tracking of the pitch angle; the tracking error is very small even when the pitch angle is near 90°. Fig. 7 shows the tracking effect of the angular rate, and Fig. 8 shows the

values of the motor speeds, which are all in a reasonable range. Because attitude values represented by a quaternion are not easily interpreted, the attitude tracking error is observed via the following method[30]

$$\xi = E(\boldsymbol{R}, \boldsymbol{R}_{\text{com}}) = 2 - \sqrt{1 + \text{tr}\left(\boldsymbol{R}_{\text{com}}^{\text{T}} \boldsymbol{R}\right)}, \tag{67}$$

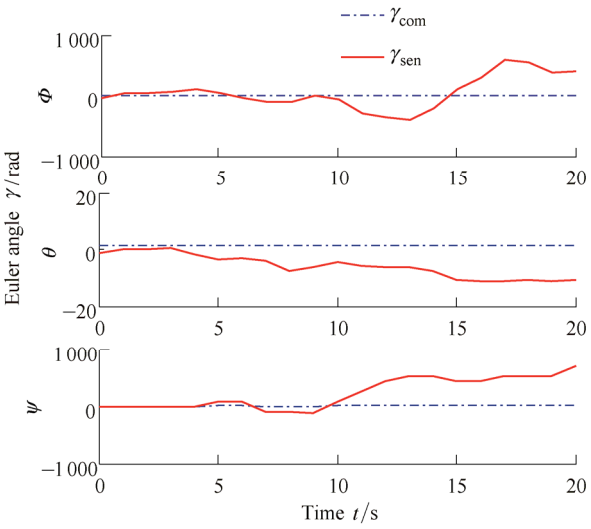

Fig. 4. Failed Euler angle tracking

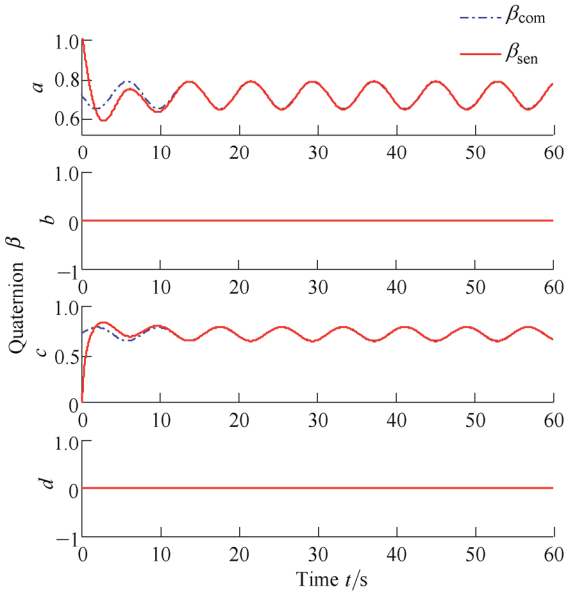

Fig. 5. Quaternion tracking in case A

where $\boldsymbol{R}$ represents the attitude matrix and is calculated by solving Eq. (58). The expression tr($\boldsymbol{X}$) represents the trace of the matrix $\boldsymbol{X}$; it can be shown that $E(\boldsymbol{R}, \boldsymbol{R}_{\text{com}}) \geqslant 0$ only when $\boldsymbol{R}=\boldsymbol{R}_{\text{com}}$, and $E=0$. It can also be shown that the value of $E(\boldsymbol{R}, \boldsymbol{R}_{\text{com}})$ increases with the attitude tracking error. The quaternion-based attitude tracking error is shown in Fig. 9. Because of the large difference between the initial attitude and the command attitude, the attitude tracking error ξ is initially large, but it converges quickly and eventually stabilizes to approximately zero, which reflects the good tracking performance and robustness of the controller. These simulation results show that the quaternion-based controller does not exhibit any singular points.

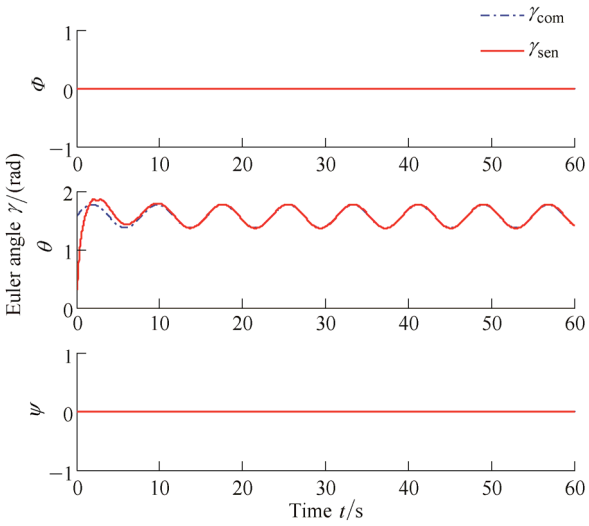

Fig. 6. Euler angle tracking in case A

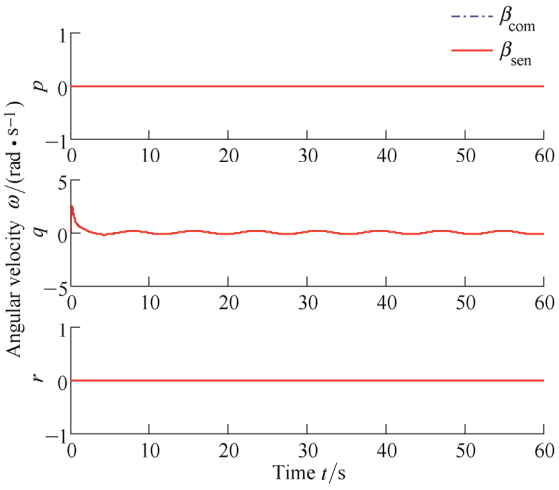

Fig. 7. Angular velocity tracking in case A

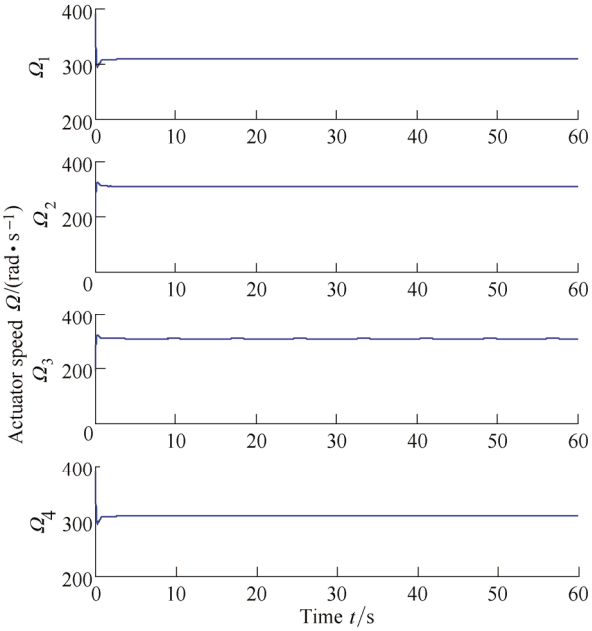

Fig. 8. Actuator speed control in case A

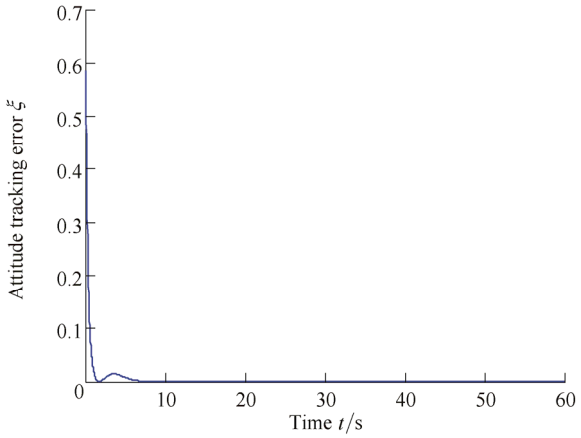

Fig. 9. Attitude tracking error in case A

4.1.2 Case B

To verify the proposed controller tracking capability and robustness properties, the input disturbances and sensor delays are added to the simulation environment. In the simulation, 1) the initial position and velocity of the quadrotor UAV are set to 0 and the attitude angle and the angular velocity are 0; 2) the external disturbance forces and moment acting on the simulation platform are shown in Fig. 10; the force and moment disturbances are pulses of 1.5 s in duration; 3) the sensor delays are shown in Table 3. The position command is given as follows: the quadrotor UAV is flying along a spiral path with a radius of 10.6 meters.

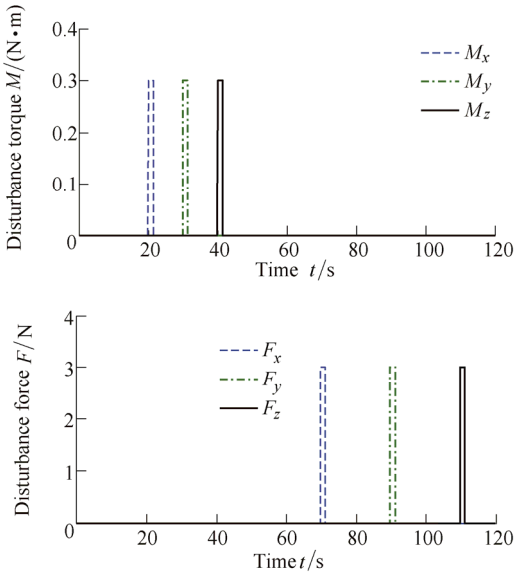

Fig. 10. Torque and force disturbance profile

Table 3. Sensor delay time

Sensor type	Delay time t/s
Position	0.5
Velocity	0.2
Angle	0.1
Angle velocity	0.02

The simulation results are shown in Figs. 11–19. It can be observed from Fig. 11 that the quadrotor UAV can achieve effective tracking of the position commands under the action of the controller. The spiral maneuver requires that the controller maintain high dynamic performance. At various interference times, there are large position deviations between the commanded position and the sensed position; however, the deviations converge quickly, which demonstrates the favorable robustness of the controller. Figs. 12–15 show the actual states of the four control loops of the controller tracking instructions, respectively. From the figures, it can be observed that the velocity, attitude and angular velocity initially exhibit large fluctuations in response to torque and force disturbances, but these fluctuations are quickly suppressed. Because of the operation of the controller, fluctuations in the position are relatively small. Figs. 16–18 show the moment, force and actuator speed commands, respectively; they are all within a reasonable range. Fig. 19 illustrates the attitude tracking error, from which we can observe that the attitude error temporarily increases in the presence of interference moment and force but soon converges to approximately zero; it also shows that the attitude tracking error is relatively small. In summary, these simulation results demonstrate that the proposed controller exhibits favorable robustness and the ability to suppress interference.

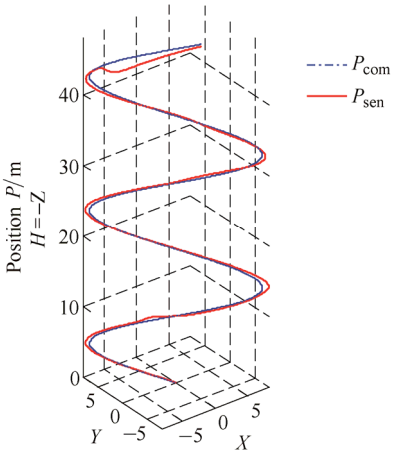

Fig. 11. 3D position tracking in case B

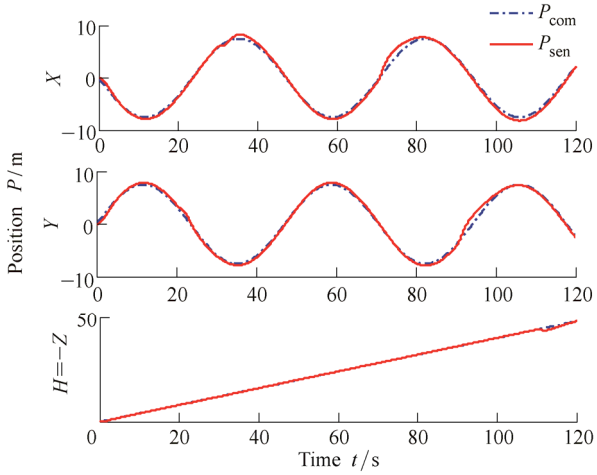

Fig. 12. Position tracking in case B

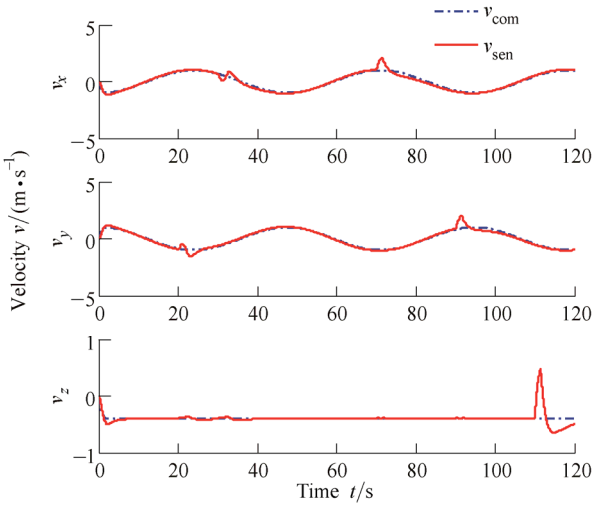

Fig. 13. Velocity tracking in case B

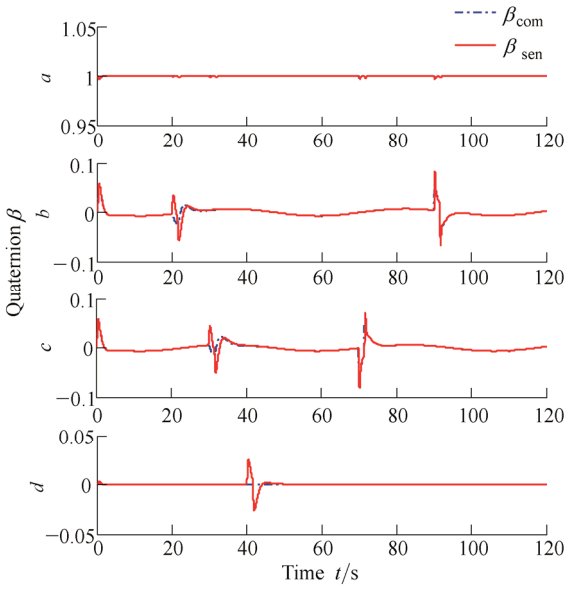

Fig. 14. Quaternion tracking in case B

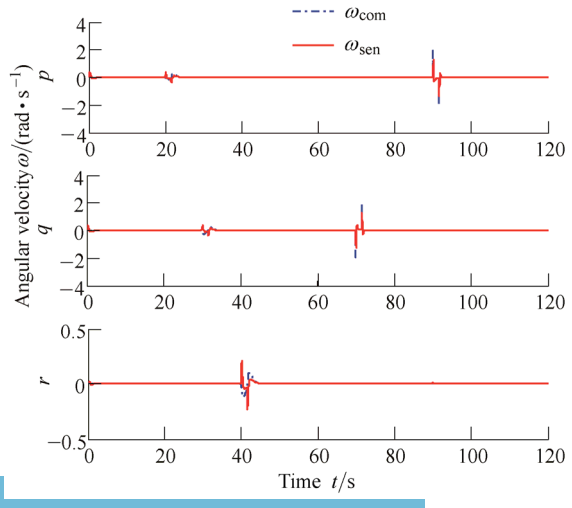

Fig. 15. Angular velocity tracking in case B

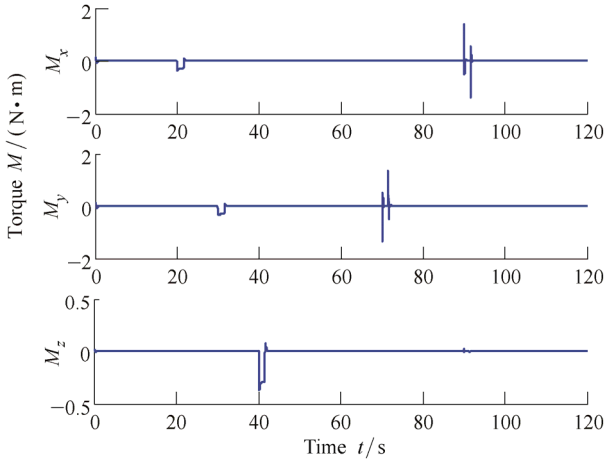

Fig. 16. Torque control in case B

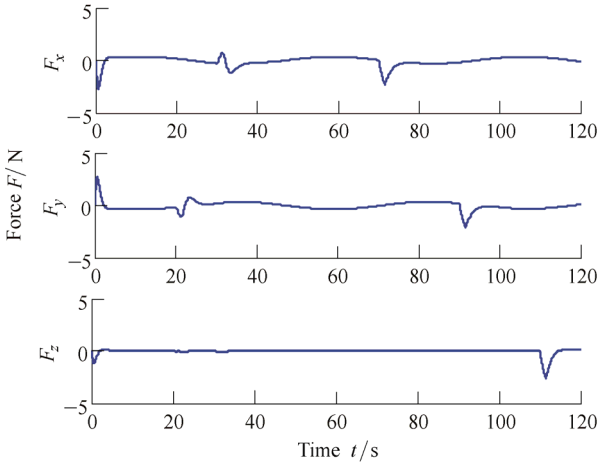

Fig. 17. Force control in case B

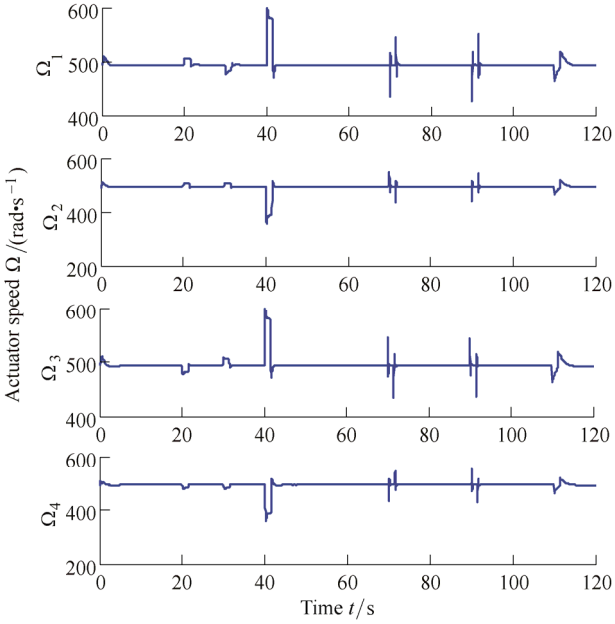

Fig. 18. Actuator speed control in case B

4.1.3 *Case C*

The control performance of the improved controller and the traditional controller are now compared. The test conditions are identical to those of case study B. The two

controllers' position, velocity, attitude, and angular velocity of the four control loop tracking errors are shown in Figs. 20–23, respectively. Among them, the dotted line and the subscript "old" represent the traditional old controller data; the solid line and the subscript "new" represent the improved new controller data.

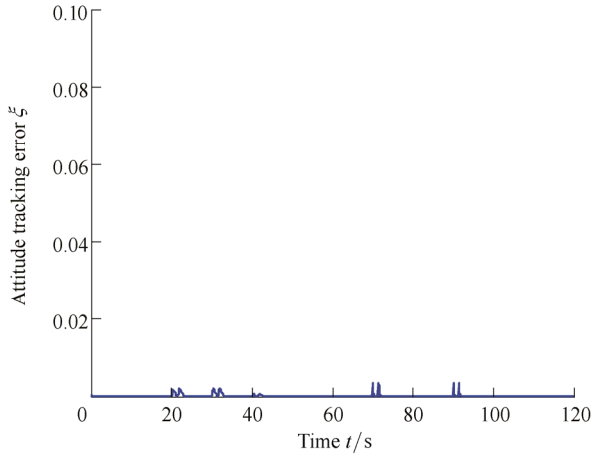

Fig. 19. Attitude tracking error in case B

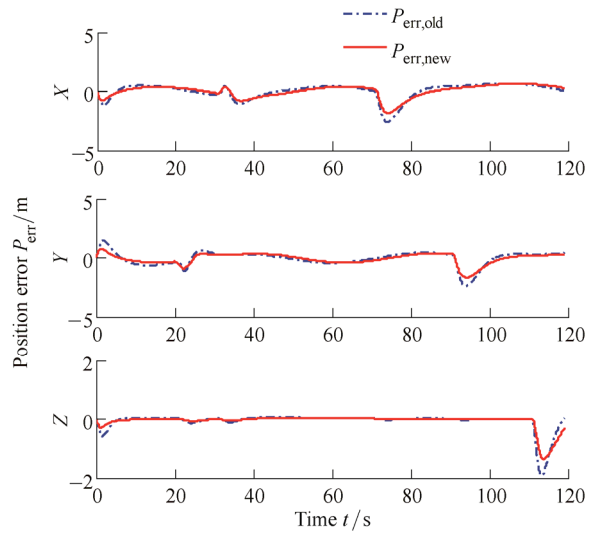

Fig. 20. Position tracking error in case C

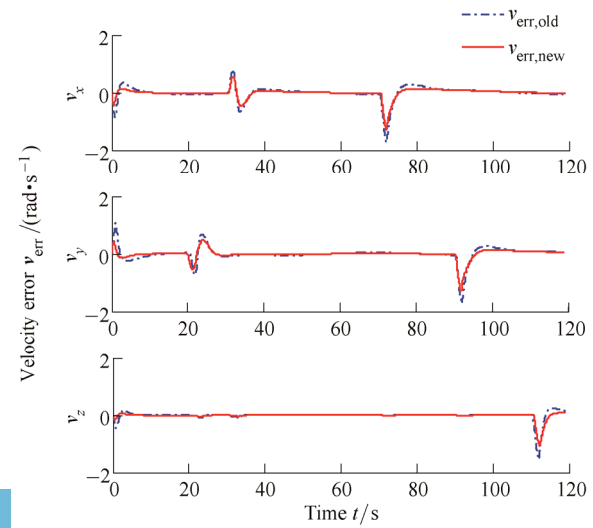

Fig. 21. Velocity tracking error in case C

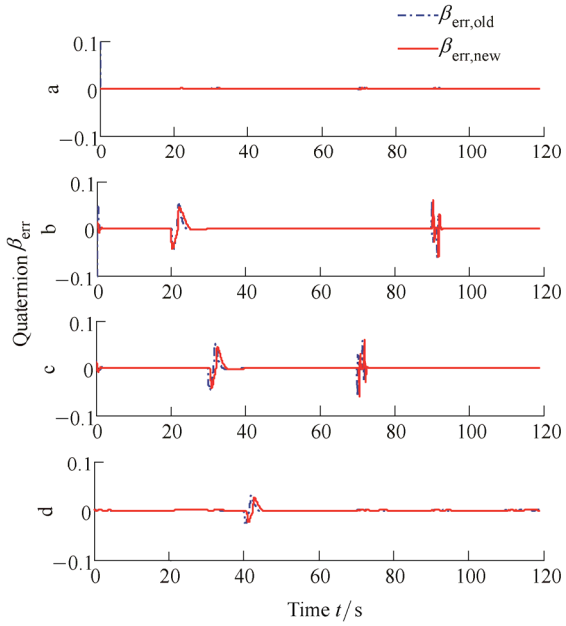

Fig. 22. Quaternion tracking error in case C

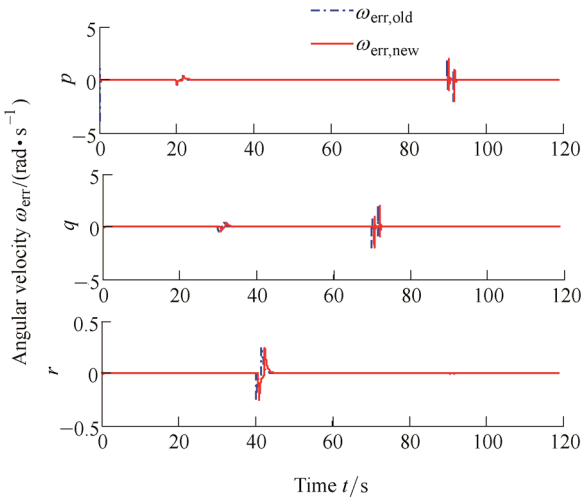

Fig. 23. Angular velocity tracking error in case C

From the test results, it can be observed that the tracking errors of all control loops of the two controllers are very close. However, in the presence of external interference, the new controller's position and speed error are smaller. The test results show that the improved controller enables both a simplified design process and certain performance advantages.

4.2 Flight results

The prototype, shown in Fig. 24, is a customized quadrotor UAV developed in-house. Its avionics consist primarily of a flight controller, an integrated navigation unit and communication systems. The integrated navigation unit obtains the actual 3D positions, velocities, attitudes and angular rates of the UAV and relays them to the flight controller. The horizontal positioning accuracy of the integrated navigation system is 2.0 m; the vertical positioning accuracy is 0.8 m. The velocity accuracy is 0.1 m/s. The roll and pitch accuracy is 0.5°, and the yaw

accuracy is approximately 1.0°.

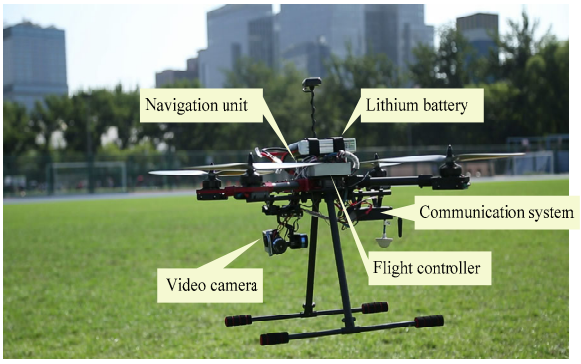

Fig. 24. Quadrotor UAV prototype

The flight controller receives commands from the communication system and calculates the control laws to drive the propellers. Fig. 25 shows the flight controller circuit board. The microcontroller used for the flight controller is a 180 MHz Cortex-M4 CPU with 260 KB RAM and 2 MB flash memory. The flight controller board includes a power supply circuit module and contains three RS232 interfaces, two CAN interfaces, one UART interface, one SBus interface and 12 PWM outputs.

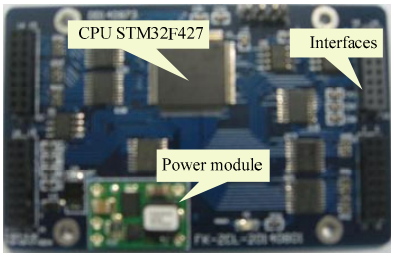

Fig. 25. Flight controller board

The test flight was performed on a sunny, breezy day ideal for controller evaluation. The test flight data in terms of position, velocity, attitude and angular velocity are shown in Figs. 26–29, respectively. From the figures, it can be observed that the controller displays good tracking performance.

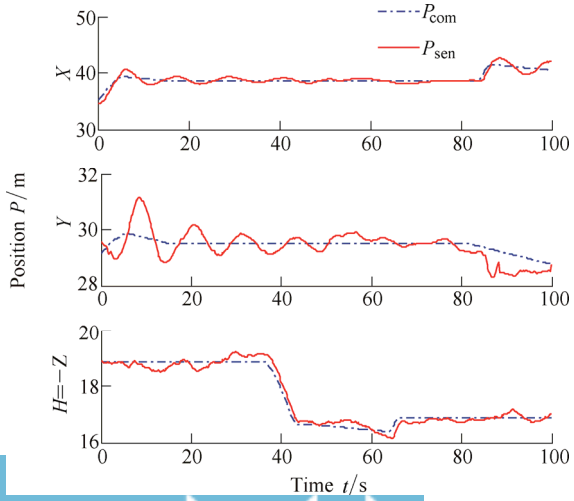

Fig. 26. Position tracking in flight

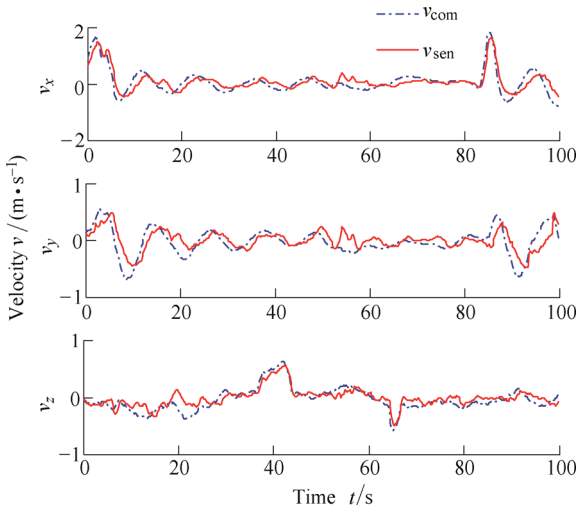

Fig. 27. Velocity tracking in flight

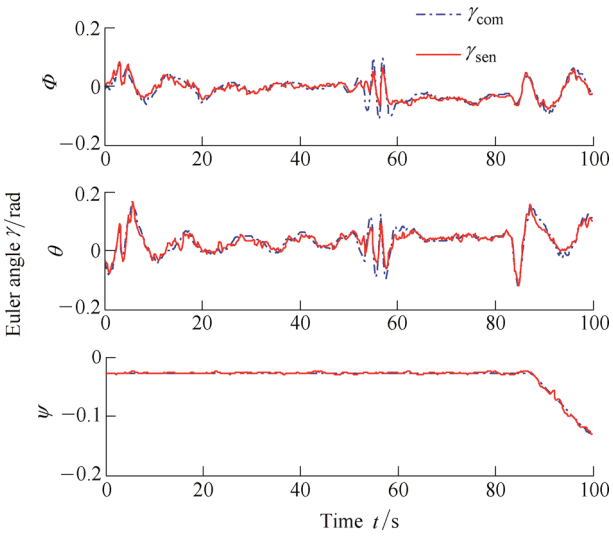

Fig. 28. Attitude tracking in flight

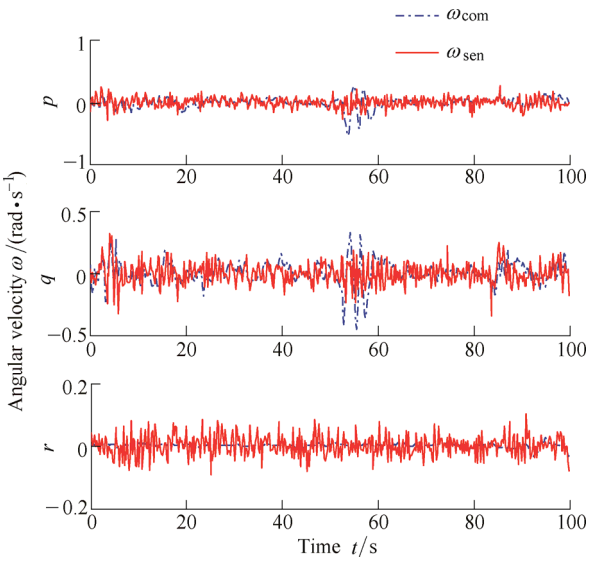

Fig. 29. Angular velocity tracking in flight

Fig. 30 shows photographs from an actual flight performed at our facility. The quadrotor UAV exhibits high stability and maneuverability.

Fig. 30. Flight tests

5 Conclusions

(1) Compared with the traditional controller based on Euler angles, the proposed quaternion-based controller avoids singularities, which is beneficial for quadrotor UAV mobility and environment interaction.

(2) Based on the quaternion attitude representation, a quadrotor UAV nonlinear controller is designed via the TLC method. The test results confirm that the proposed controller is robust.

(3) Based on the TLC theory, the control structure of the UAV controller is improved through the use of the command value, corrected by obeying the tracking error control law instead of the nominal value to design the inner loop controller. The test results demonstrate that the improved controller offers both a simplified design process and certain performance advantages.

References

[1] GOODARZI F, LEE D, LEE T. Geometric nonlinear PID control of a quadrotor UAV on SE(3)[C]//*Proceeding of European Control Conference*, Zurich, Switzerland, July 7–19, 2013: 3845–3850.

[2] BOLANDI H, REZAEI M, MOHSENIPOUR R, et al. Attitude control of a quadrotor with optimized PID controller[J]. *Intelligent Control & Automation*, 2013, 4(3): 342–349.

[3] LUQUE-VEGA L, CASTILLO-TOLEDO B, LOUKIANOV A G. Robust block second order sliding mode control for a quadrotor[J]. *Journal of the Franklin Institute*, 2012, 349(2): 719–739.

[4] ZHENG E H, XIONG J J, LUO J L. Second order sliding mode control for a quadrotor UAV[J]. *Isa Transactions*, 2014, 53(4): 1350–1356.

[5] PANOMRATTANARUG B, HIGUCHI K, MORA-CAMINO F. Attitude control of a quadrotor aircraft using LQR state feedback controller with full order state observer[C]//*SICE Annual Conference, 2013 Proceedings of*, Nagoya, Japan, September 14–17, 2013: 2041–2046.

[6] ZHAO S L, AN H L, ZHANG D B, et al. A new feedback linearization LQR control for attitude of quadrotor[C]//*2014 13th International Conference on Control Automation Robotics & Vision(ICARCV)*, Singapore, December 10–12, 2014: 1593–1597.

[7] ZHENG F, GAO W N. Adaptive integral backstepping control of a micro-quadrotor[C]//*International Conference on Intelligent Control and Information Processing,* Harbin, China, July 25–28, 2011: 910–915.

[8] DAS A, LEWIS F, SUBBARAO K. Backstepping approach for controlling a quadrotor using Lagrange form dynamics[J]. *Journal of Intelligent & Robotic Systems,* 2009, 56(1): 127–151.

[9] HAN J D, ZHU Z Q, JIANG Z Y, et al. Simple PID parameter tuning method based on outputs of the closed loop system[J]. *Chinese Journal of Mechanical Engineering*, 2016, 29(3): 465–474.

[10] REN H P, FAN J T. Adaptive backstepping slide mode control of pneumatic position servo system[J]. *Chinese Journal of Mechanical Engineering*, 2016, 29(6): 1003–1009.

[11] MELLINGER D, MICHAEL N, KUMAR V. Trajectory generation and control for precise aggressive maneuvers with quadrotors[J]. *International Journal of Robotics Research,* 2012, 31(5): 664–674.

[12] MELLINGER D, KUMAR V. Minimum snap trajectory generation and control for quad rotors[C]//*2011 IEEE International Conference on Robotics and Automation(ICRA),* Shanghai, China, May 9–13, 2011: 2520-2525.

[13] YU Y S, DING X L. On hybrid modeling and control of a multi-propeller multifunction aerial robot with flying-walking locomotion[J]. *Autonomous Robots*, 2015, 38(3): 225–242.

[14] DING X L, YU Y S, ZHU J J. Trajectory linearization tracking control for dynamics of a multi-propeller and multifunction aerial robot-MMAR[C]//*Proceedings of IEEE International Conference on Robotics and Automation*, Shanghai, China, May 9–13, 2011: 757– 762.

[15] PARK J. Interpolation and tracking of rigid body orientations[C]// *2010 International Conference on Control Automation and System*(*ICCAS),* Gyeonggi-do, October 27–30, 2010: 668–673.

[16] LI J H, P W. Smooth interpolation on homogeneous matrix groups for computer animation[J]. *Journal of Zhejiang University Science A(Science in Engineering)*, 2006, 7(7): 1168–1177.

[17] YU Y S, DING X L, ZHU J J. Attitude tracking control of a quadrotor UAV in the exponential coordinates[J]. *Journal of the Franklin Institute*, 2013, 350(8): 2044–2068.

[18] LIU H, WANG X, ZHONG Y. Quaternion-based robust attitude control for uncertain robotic quadrotors[J]. *IEEE Transactions on Industrial Informatics,* 2015, 11(2): 406–415.

[19] ZHA C L, DING X L, YU Y S, et al. Design and implementation of a compact integrated navigation system for micro multi-propeller multifunction aerial robots[J]. *Applied Mechanics & Materials,* 2014, 556–562: 1553–1559.

[20] HUANG R, LIU Y, ZHU J J. Guidance, navigation, and control system design for tripropeller vertical-takeoff-and-landing unmanned air vehicle[J]. *Journal of Aircraft*, 2009, 46(6): 1837–1856.

[21] TITTERTON D H, WESTON J L. *Strapdown inertial navigation technology*[M]. 2nd ed. London, United Kingdom: The Institution of Electrical Engineer, 2004.

[22] LI B K, CAO Y, ZHANG Q J, et al. Orientation-singularity representation and qrientation-capability computation of a special class of the gough-stewart parallel mechanisms using unit quaternion[J]. *Chinese Journal of Mechanical Engineering*, 2012, 25(6): 1096–1104.

[23] WU X F, IGNATOV R, MUENST G, et al. A nonlinear flight controller design for a UFO by trajectory linearization method. I. modeling[C]//*System Theory, 2002. Proceedings of the Thirty-Fourth Southeastern Symposium on*, Huntsville, Alabama, March 18–19, 2002: 97–102.

[24] TAYEBI A, MCGILVRAY S. Attitude stabilization of a VTOL quadrotor aircraft[J]. *IEEE Transactions on Control Systems Technology*, 2006, 14(3): 562–571.

[25] MAHONY R, HAMEL T. Adaptive compensation of aerodynamic effects during takeoff and landing manoeuvres for a scale model autonomous helicopter[J]. *European Journal of Control,* 2001, 7(1): 43–57.
[26] ZHU J J. PD-spectral theory for multivariable linear time-varying systems[C]//*Proceedings of IEEE Conference on Decision and Control*, San Diego, CA, December 10–12, 1997: 3908–3913.
[27] WU X F, ADAMI T, CAMPBELL J, et al. A nonlinear flight controller design for a UFO by trajectory linearization method. II. controller design[C]//*System Theory, 2002. Proceedings of the Thirty-Fourth Southeastern Symposium on*, Huntsville, Alabama, March 18– 19, 2002: 103–107.
[28] LIU Y, ZHU J J, WILLIAMS R L, et al. Omni-directional mobile robot controller based on trajectory linearization[J]. *Robotics and Autonomous Systems*, 2008, 56(5): 461–479.
[29] SHEPPARD, S W, Quaternion from rotation Matrix[J]. *Journal of Guidance and Control*, 1978, 1(3): 223–224.
[30] LEE T. Geometric tracking control of the attitude dynamics of a rigid body on SO(3)[C]//*Proceedings of American Control Conference*. San Francisco, CA, June 29–July 1, 2011: 1200–1205.

Biographical notes

ZHA Changliu, born in 1975, is currently a PhD candidate at *School of Mechanical Engineering and Automation, Beihang University, China.* His research interests include navigation and control of robotic systems, in particular, aerial robots.
E-mail: zhachangliu@ buaa.edu.cn

DING Xilun, born in 1967, is currently a professor and a PhD candidate supervisor at *School of Mechanical Engineering and Automation, Beihang University, China*. He received his PhD degree from *Harbin Institute of Technology, China*, in 1997. His research interests include the dynamics of compliant mechanical systems and robots, nonholonomic control of space robots, dynamics and control of aerial robots, and biomimetic robots.
E-mail: xlding@buaa.edu.cn

YU Yushu, born in 1985, is currently a postdoctoral fellow at *School of Mechanical Engineering and Automation, Beihang University, China*. His research interests include dynamics and control of robotic systems, in particular, aerial robots.
E-mail: yushuyu@buaa.edu.cn

WANG Xueqiang, born in 1988, is currently a PhD candidate at *School of Mechanical Engineering and Automation, Beihang University, China*. His research interests include dynamics and control of robotic systems, in particular, aerial robots.
E-mail: wangxueqiang0922@163.com